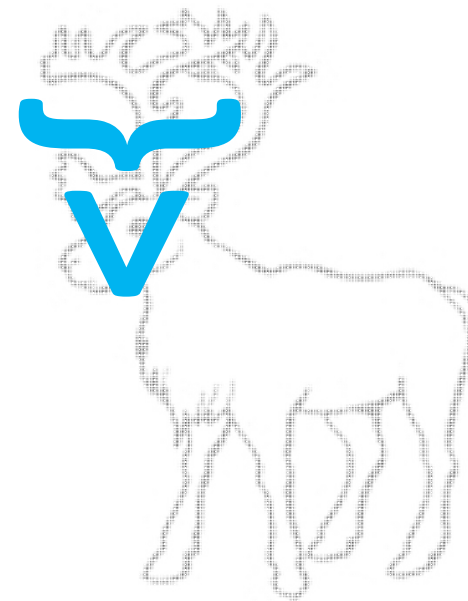
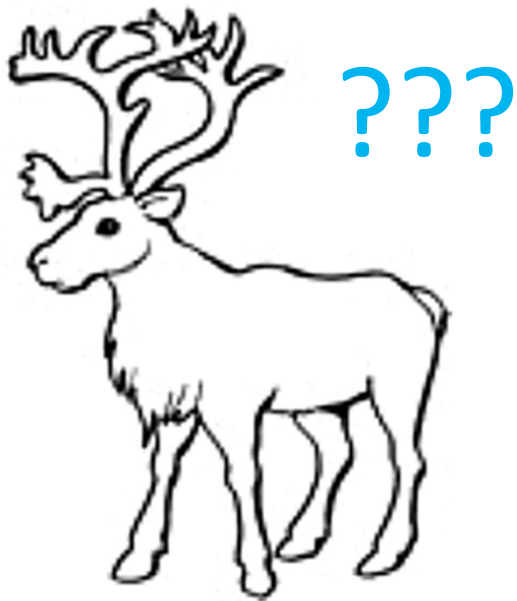


Moderne Web-Anwendungen mit Vaadin

Sebastian.Rothbucher@akquinet.de

September 2013





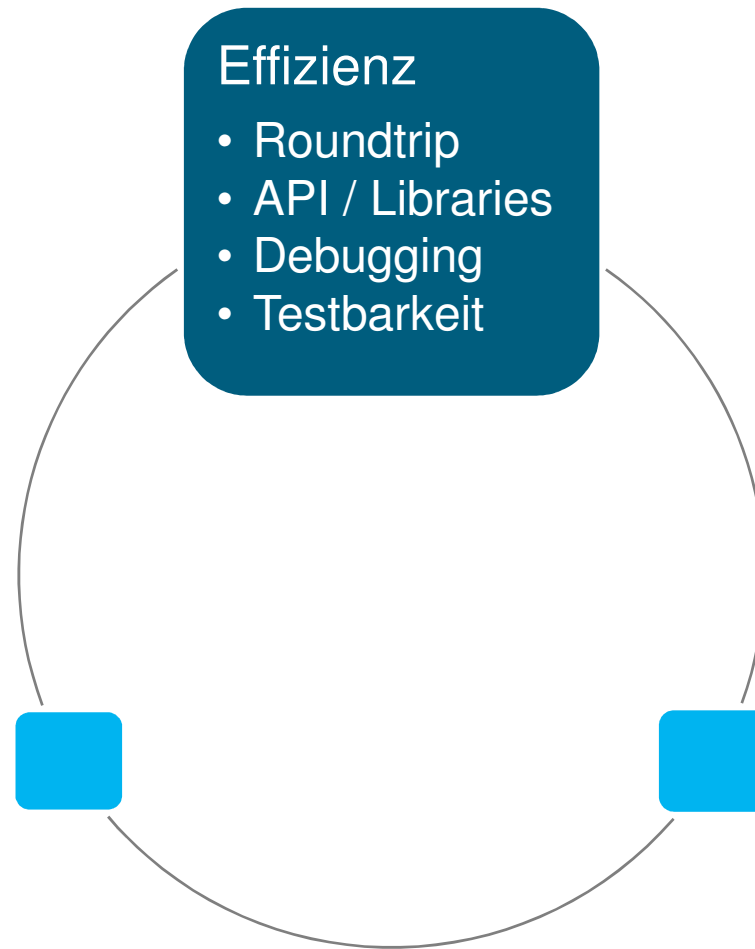
vaadin }> Solution Partner

- **Was** sind tägliche Herausforderungen?
- **Warum** ist Vaadin bei deren Bewältigung hilfreich?
- **Wie** wird in Vaadin eine Anwendung entwickelt?
- **Womit** anfangen?

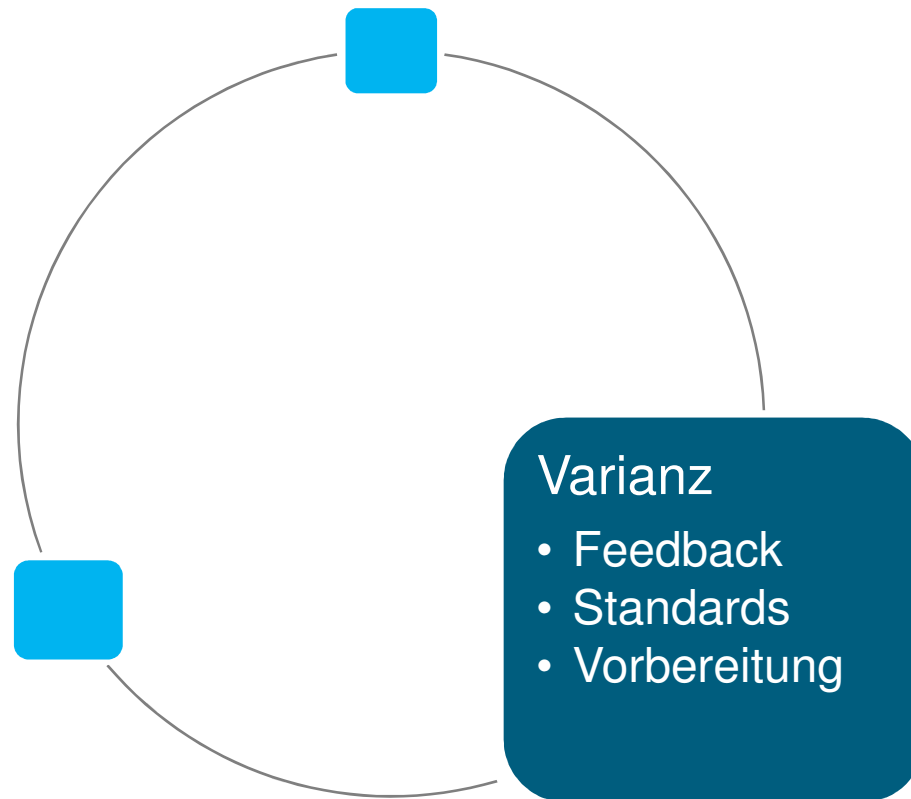
- **Was** sind tägliche Herausforderungen?
- **Warum** ist Vaadin bei deren Bewältigung hilfreich?
- **Wie** wird in Vaadin eine Anwendung entwickelt?
- **Womit** anfangen?



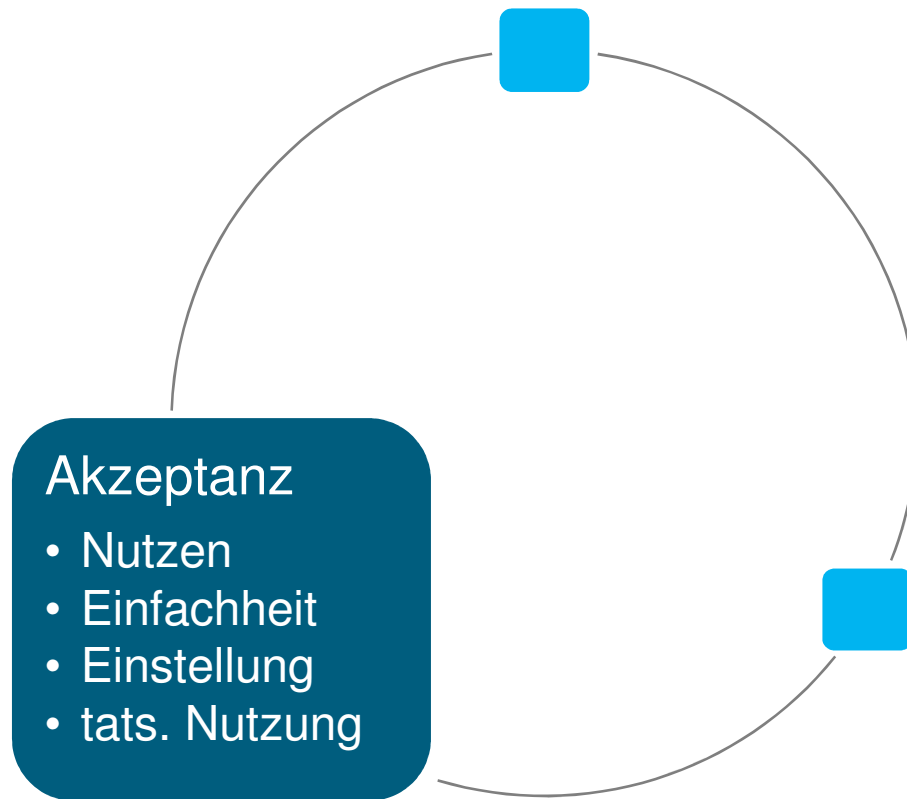
Was sind tägliche Herausforderungen?



Was sind tägliche Herausforderungen?

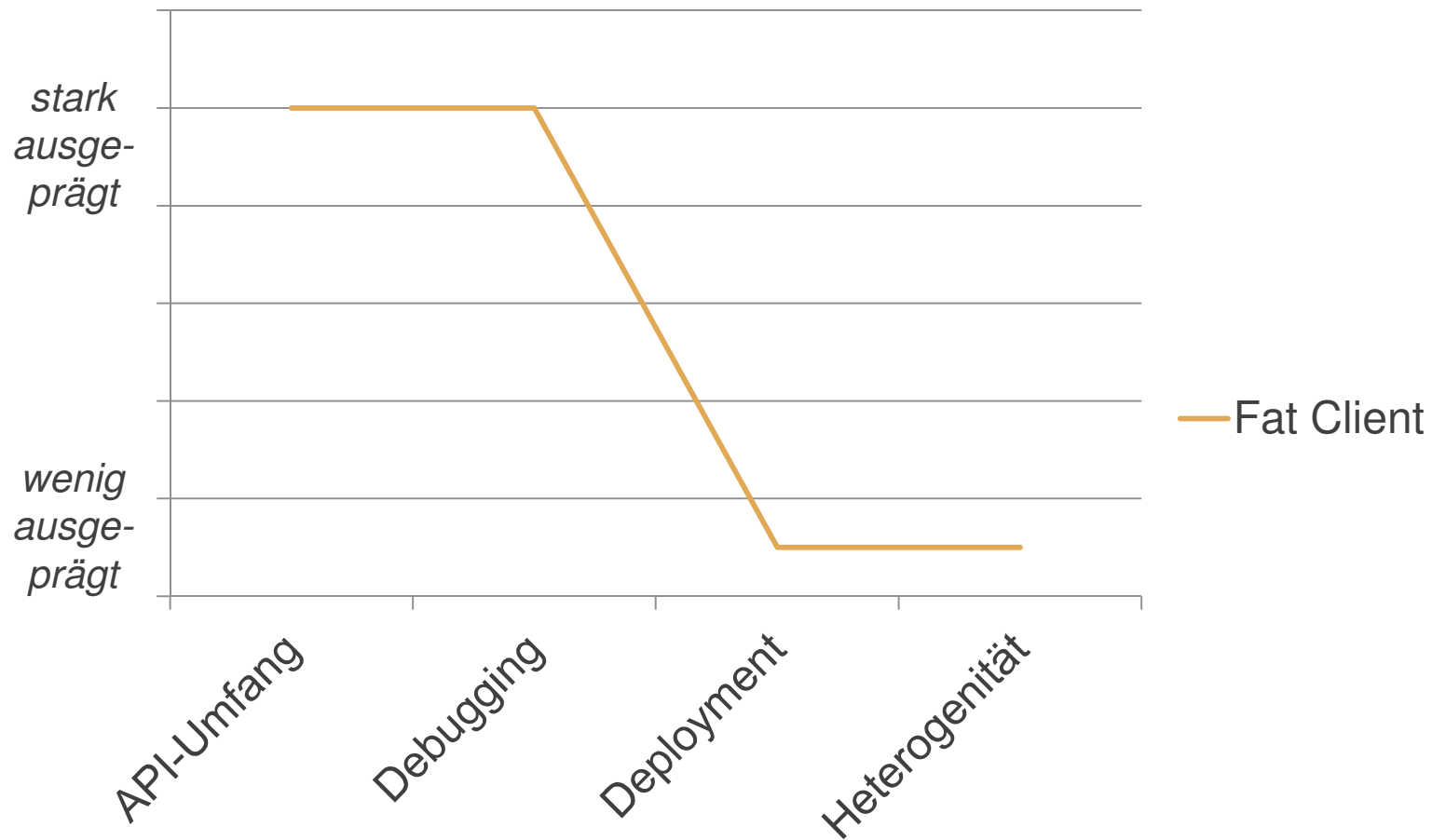


Was sind tägliche Herausforderungen?

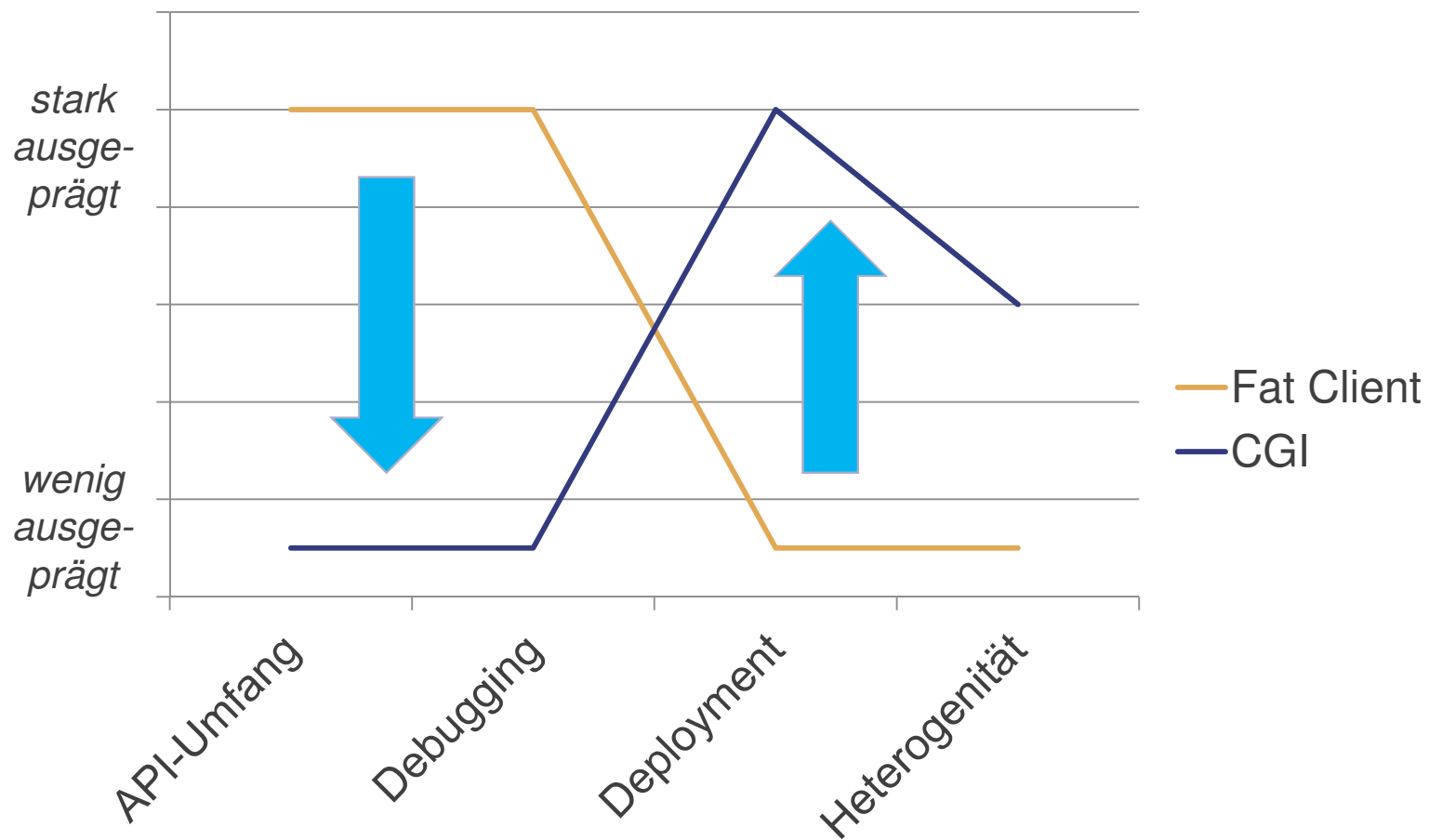


- Was sind tägliche Herausforderungen?
- **Warum** ist Vaadin bei deren Bewältigung hilfreich? ←
- Wie wird in Vaadin eine Anwendung entwickelt?
- Womit anfangen?

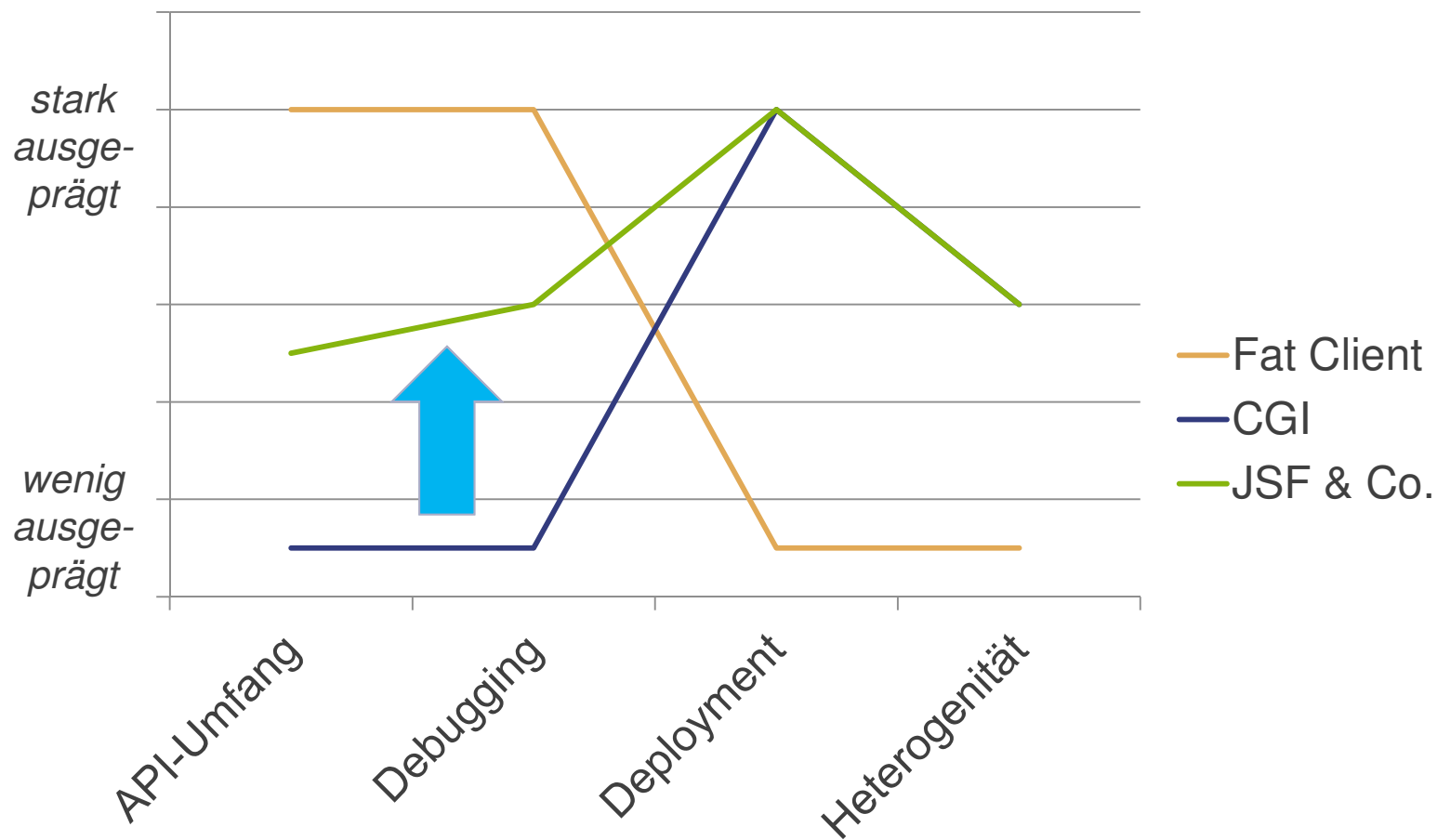
Wege zu Vaadin – Fat Client (Ausgangspunkt)



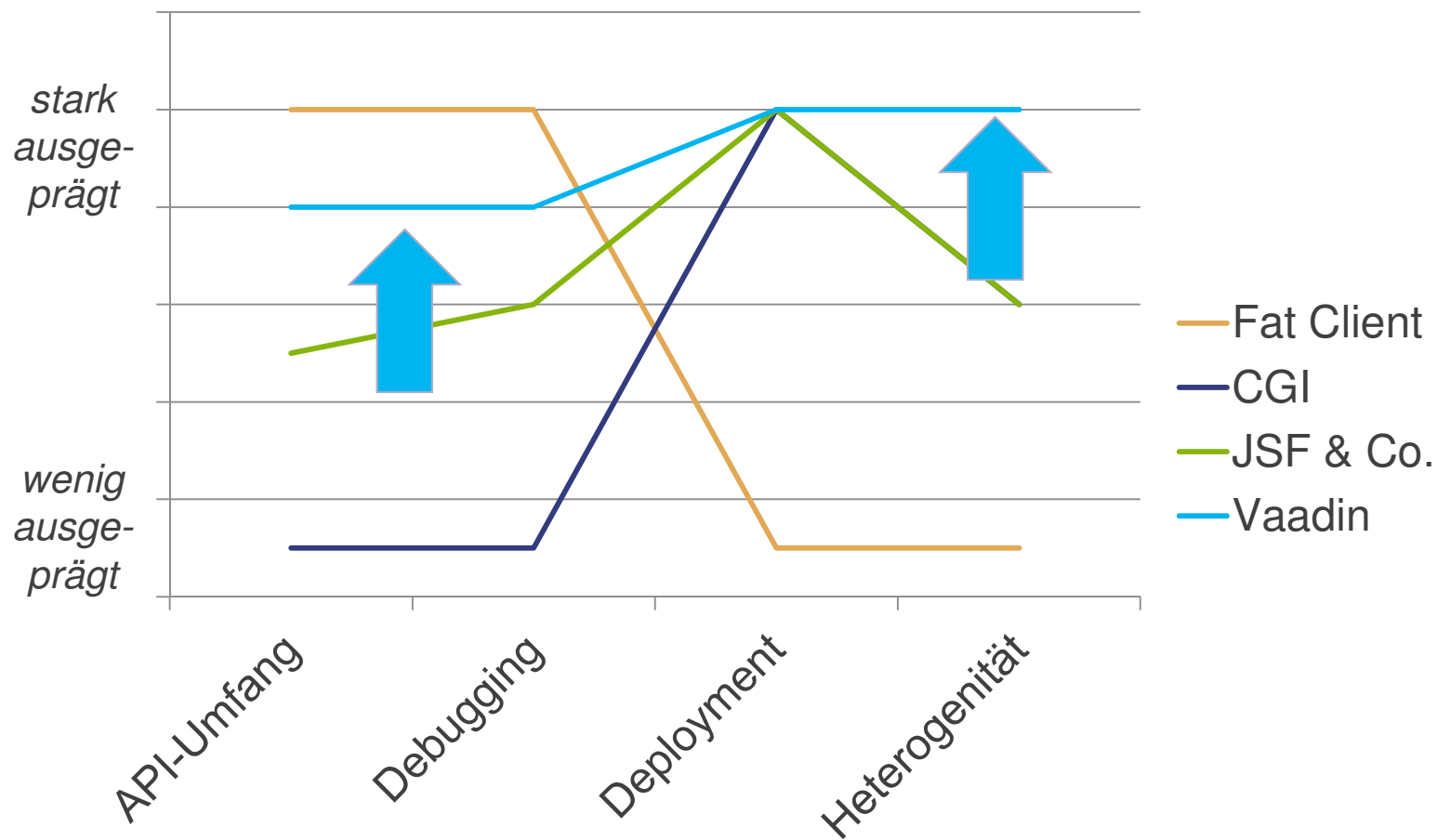
➔ maximaler Umfang für einen kleinen Kreis



➔ sehr rudimentäre, aber eben weithin einsetzbare Architektur



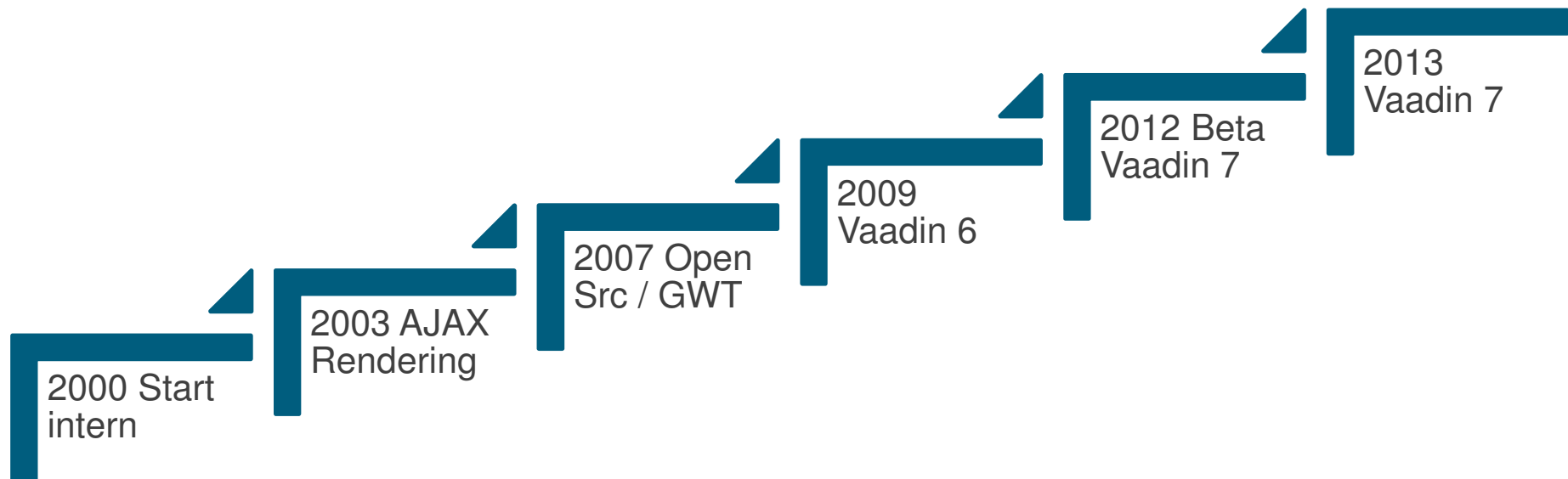
➔ in der Effizienz verbesserte, nach wie vor weithin einsetzbare Architektur



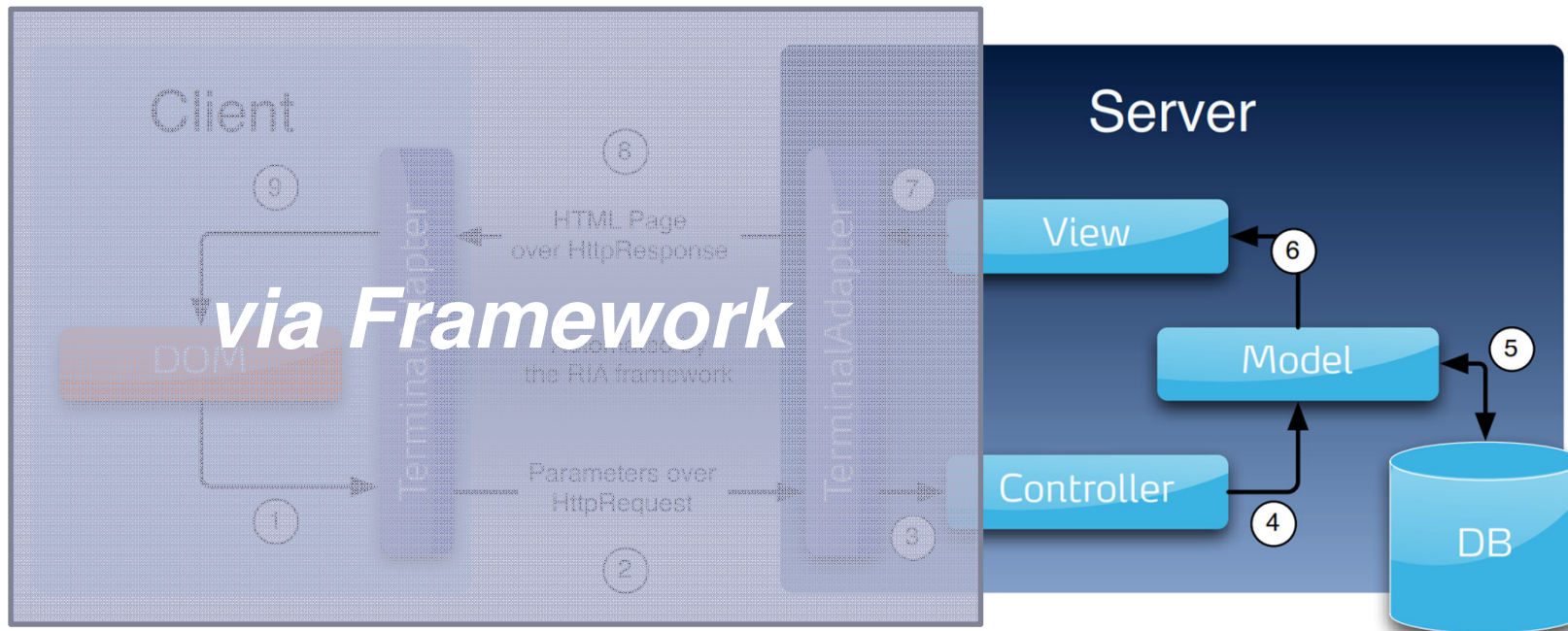
➔ Evolution zu Perfektion: sehr großer Umfang für eine sehr breite Zielgruppe

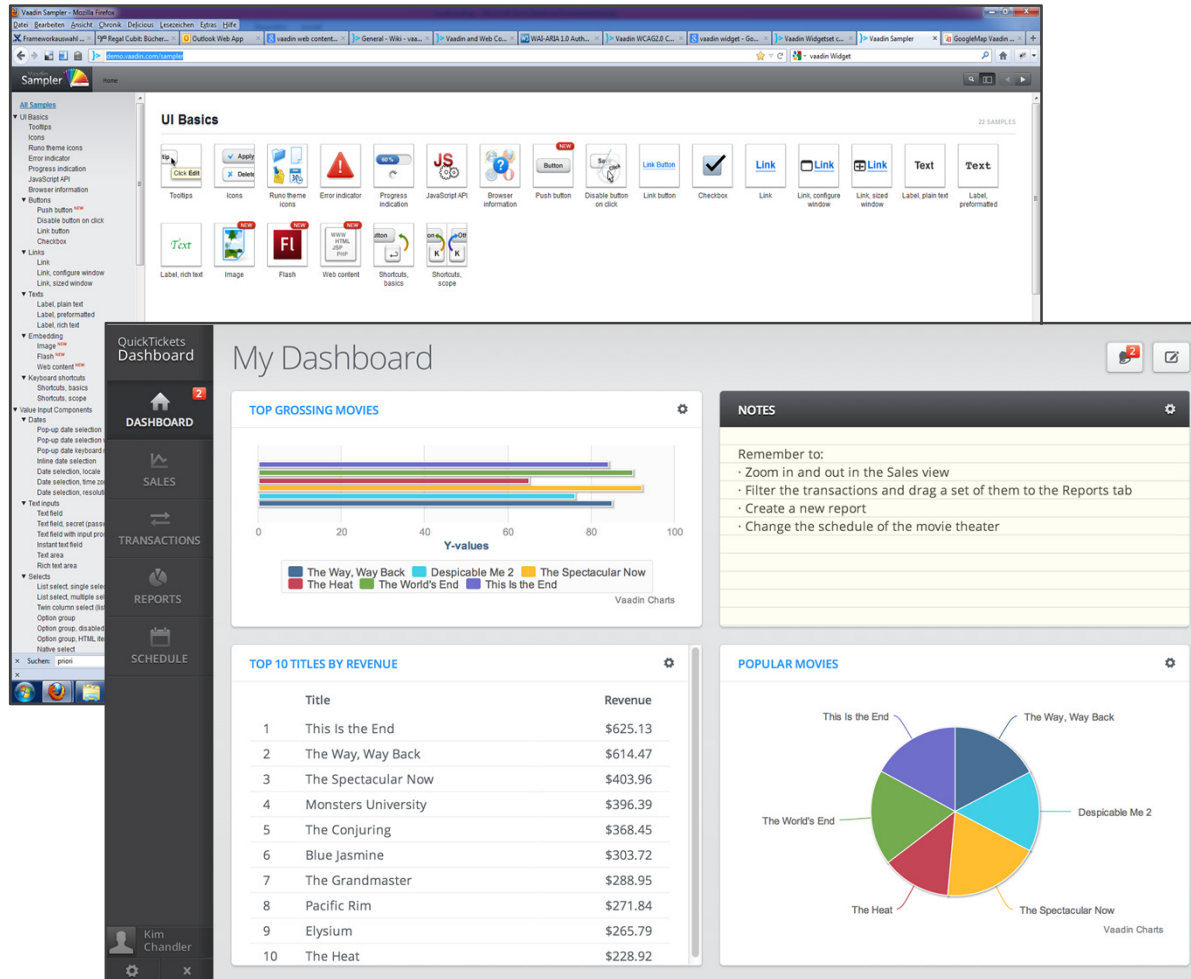
- **Rich Internet Applications** = RIA (im Browser)
- Auf **Desktop-Niveau** für die Anwender
- **Mittels Komponenten** (wie SWT / Swing) für die Entwickler
- In **pure Java** entwickeln

➔ *Streng typisierte Web-Entwicklung*



➔ *Stabilität + Aktualität*

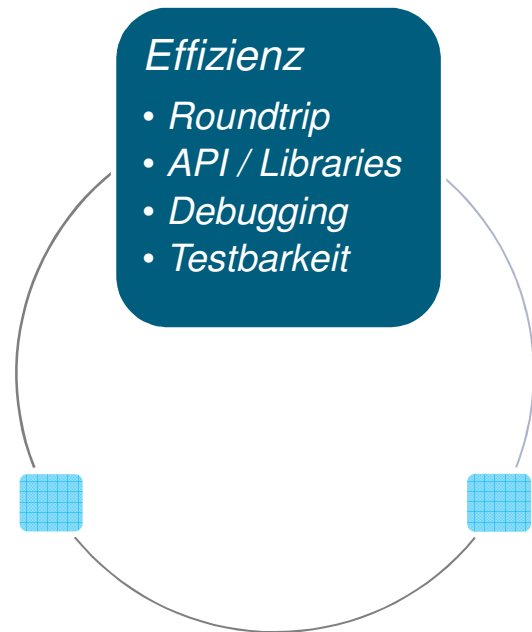




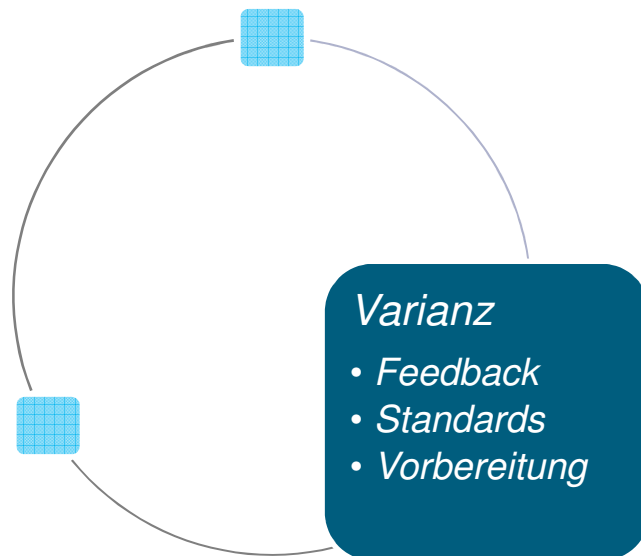
- **45+ Komponenten**
300+ im Add-on Dir
- **5+ Layouts**
- **3 Themes**
1 weitere in Arbeit
- **Drag & Drop Support**
- **Touch Events**
- **CSS + SASS**



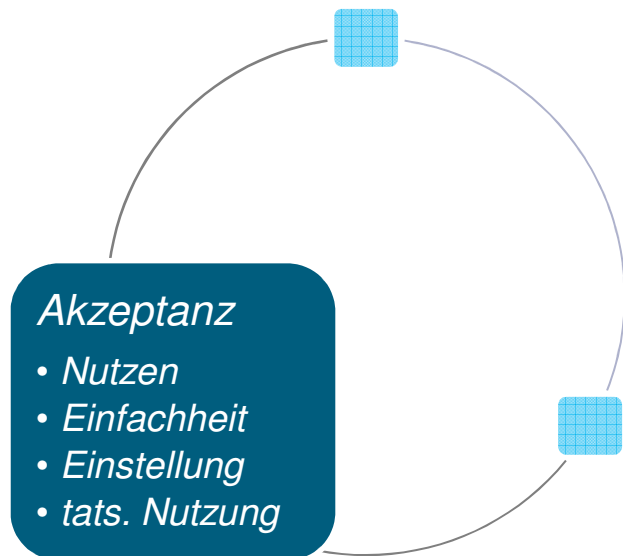
- Nach wie vor RIA
- Nach wie vor Pure Java
- iOS Look & Feel
Neutraler Look möglich
- Navigationsmodell
Listen scrollen,
Navigation Bar



- Maven, Hot Deployment, JRebel
- UI Komponenten Sammlung
- Full Java
- Remote Debugging
- JUnit
- Selenium
- JMeter




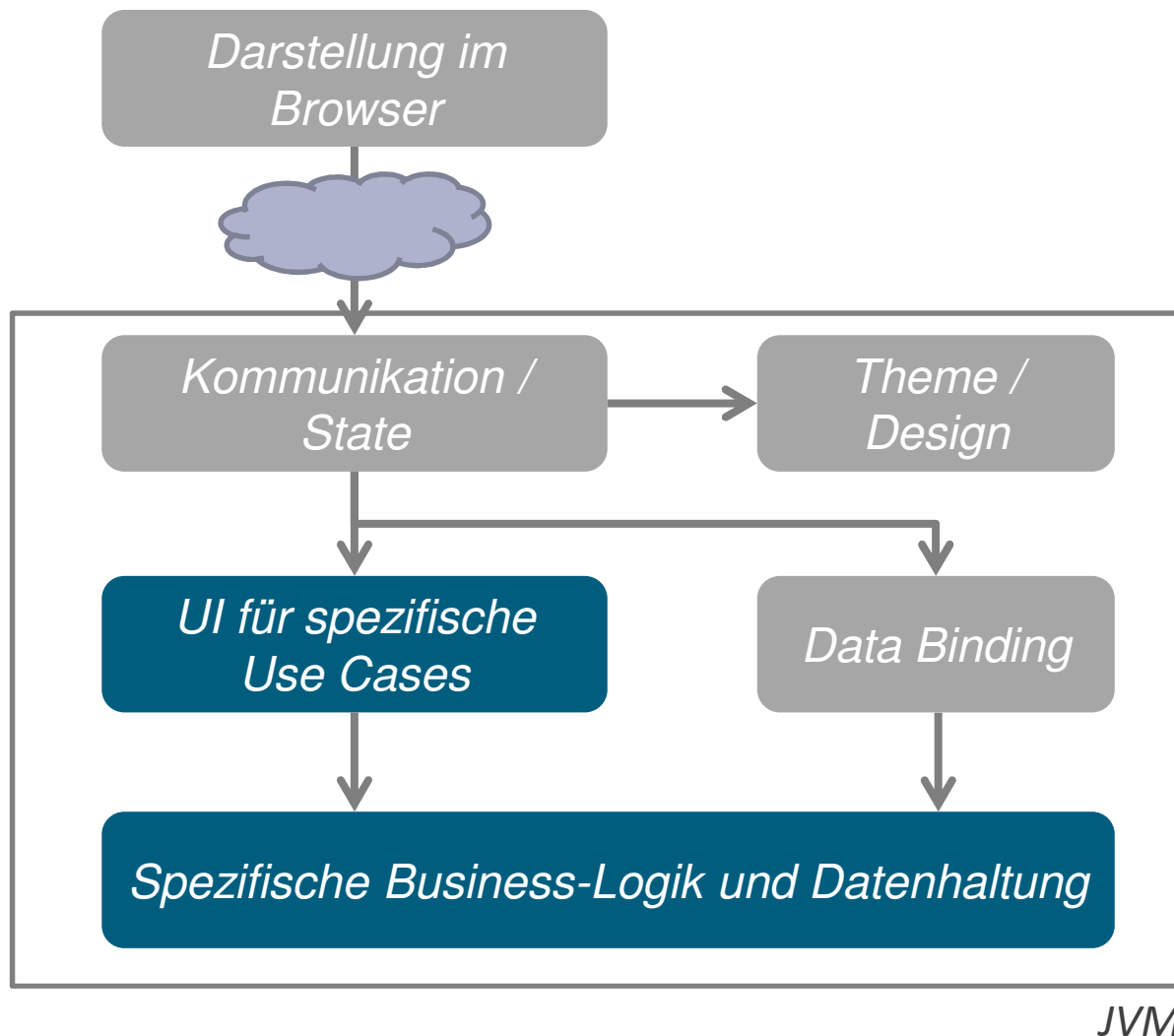
- Schnelle erste Version
- Qualifiziertes Feedback
- Standard JEE-Technologie
- Keine Kinderkrankheiten
- Gehärtet
- Große UI Komponenten Sammlung



- Erwartungskonforme UI
- Style und Usability
- Mobile

- Zeit für Business-Logik
- Schnelles Feedback

- Was sind tägliche Herausforderungen?
- Warum ist Vaadin bei deren Bewältigung hilfreich?
- **Wie wird in Vaadin eine Anwendung entwickelt?** 
- Womit anfangen?



- Client State im Hintergrund abgewickelt
- Erstellen von Widgets (analog Swing / SWT) auf dem Server
- Verwenden von Web / JEE Container
- Fokus auf spezifische Logik (türkis)

Sprung in den Code: *init()*



```
@Theme("officewerker")
public class OfficewerkerApplicationUI extends UI {

    @Inject
    Notificator notificator;

    ...

    @Override
    public void init(VaadinRequest request) {

        this.i18nLocaleConfig.setLocale(getLocale());

        ...

        setContent((ComponentContainer) this.mainPresenter
            .getView().getComponent());

        Page.current().setTitle("Officewerker");
    }
}
```

← Eigener Theme

← main() von Vaadin

← Fenster analog Swing

← MVP ist eine Option

```
public void init() {  
    setSizeFull();
```

```
    Toolbar toolbar = new Toolbar(  
        this.i18n.get("login.title"), null);
```

← Internationalisierung

```
    this.textFieldMandator = new TextField(  
        this.i18n.get("login.label.mandator"));
```

← TextField analog Swing

```
    this.textFieldMandator.setWidth("100%");
```

```
    this.textFieldMandator.setTabIndex(1);
```

```
    this.textFieldMandator.focus();
```

```
    this.textFieldMandator.setId("loginMandant");
```

```
    this.textFieldMandator.addStyleName("styleid-loginMandant");
```

← Vorbereitung
Oberflächentest

```
    this.textFieldLogin = new TextField(  
        this.i18n.get("login.label.login"));
```

```
    this.textFieldLogin.setWidth("100%");
```

```
    this.textFieldLogin.setTabIndex(2);
```

```
    this.textFieldLogin.setDebugId("loginName");
```

```
    this.textFieldLogin.addStyleName("styleid-loginName");
```

```
    HorizontalLayout layoutLoginMandator = new HorizontalLayout();
```

← Panels analog Swing

```
    layoutLoginMandator.setWidth("100%");
```

```
    layoutLoginMandator.setSpacing(true);
```

```
    layoutLoginMandator.addComponent(this.textFieldMandator);
```

```
    layoutLoginMandator.addComponent(this.textFieldLogin);
```

Sprung in den Code: *init()* + *LoginComponent*



Sprung in den Code: *init()* + LoginComponent – HTML



officewerker.net

Anmeldung

Mandant Benutzer

input#loginMandant.v-textfield.v-textfield-styleid-loginMandant.styl... 134px x 24px

Anmelden

Haftungsausschluss Impressum

Elements Resources Network Sources Timeline Profiles Audits Console

```
width: 274px; height: 39px;"/>

Mandant


```

Sprung in den Code: *init()* + *LoginComponent* – HTML



The screenshot shows a web browser window with the URL `localhost:8080/officewerker-suite-war/`. The page title is "officewerker.net" and the main heading is "Anmeldung". The login form contains two text input fields labeled "Mandant" and "Benutzer", and a button labeled "Anmelden". The browser's developer tools are open, showing the DOM tree and a list of Selenium commands. The commands are:

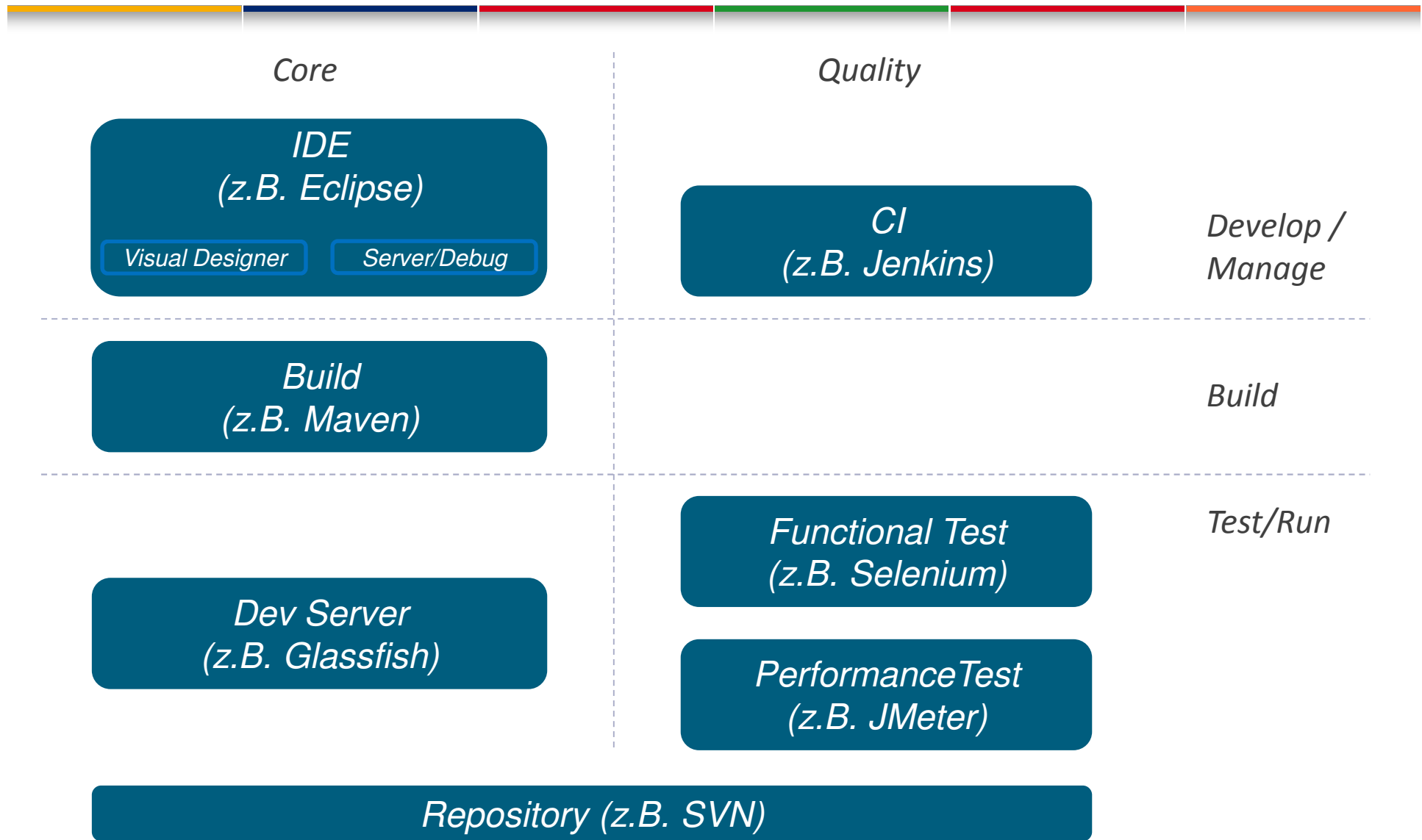
```
waitForVisible //div[contains(@class, 'styleid-HeaderComponent-buttonIngoingInvoices')]
clickAt //div[contains(@class, 'styleid-HeaderComponent-buttonIngoingInvoices')] 5, 5
waitForVisible //div[contains(@class, 'styleid-IngoingInvoiceToolBarComponent')]
clickAt //div[contains(@class, 'styleid-IngoingInvoiceToolBarComponent')] 5, 5
```

Das Zusammens...

```
waitForVisible //div[contains(@class, 'styleid-HeaderComponent-buttonIngoingInvoices')]
clickAt //div[contains(@class, 'styleid-HeaderComponent-buttonIngoingInvoices')] 5, 5
waitForVisible //div[contains(@class, 'styleid-IngoingInvoiceToolBarComponent')]
clickAt //div[contains(@class, 'styleid-IngoingInvoiceToolBarComponent')] 5, 5
```

der nächste Pun...

Command	clickAt
Target	//div[contains(@class, 'styleid-HeaderComponent-buttonIngoingInvoices')]



- EC² x-large
- zufällig 10 (dann 500) von 10k Rechnungen
- bis 200 parallele (leicht versetzte) Threads
- JMeter / JVisualVM

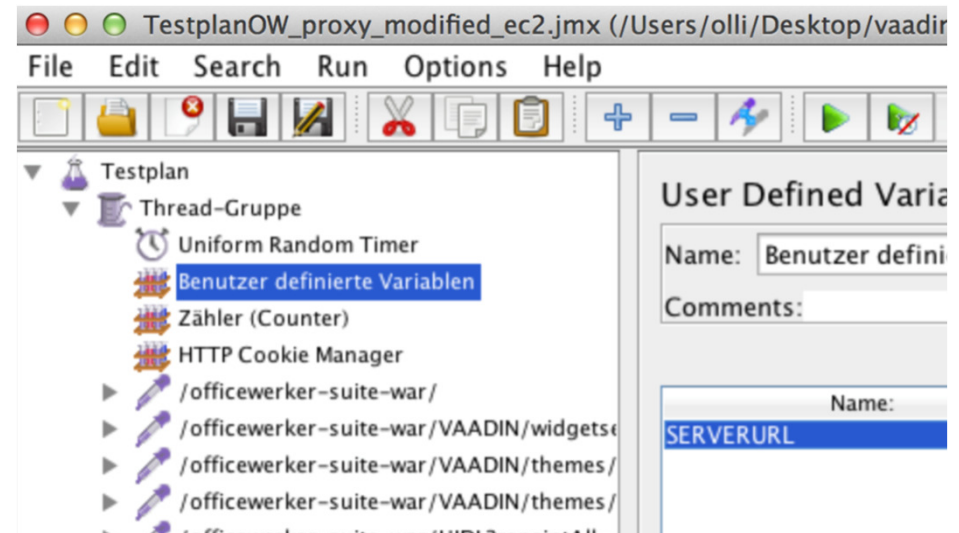
Highlights Antwortzeit (Self time):

- ~70ms je Rechnung im Durchschnitt
- 3 von 4 in weniger als ~150ms
- >50% ist DB, 98% Backend-related
- Vaadin ist nicht mehr messbar (0%)

Highlights Speicherverbrauch:

- 900k Instanzen Rechnungs-Bean
- 2.800 Instanzen „teuerste“ Vaadin-Klasse

➔ (Rework im) Backend bestimmt Performance



Siehe auch <http://blog-de.akquinet.de/2013/07/02/vaadin-unter-last-teil-1-aufsetzen-der-testumgebung/> etc.

- **Aufruf von beliebigem JavaScript** (massiv verbessert in Vaadin 7)
vaadin.forceSync() = JS 2 vaadin (Vaadin 6)
JavaScript.getCurrent().addFunction = JS 2 vaadin (Vaadin 7)
getWindow().executeJavaScript = vaadin 2 JS (Vaadin 6)
JavaScript.getCurrent().execute = vaadin 2 JS (Vaadin 7)

- **Einschließlich Integration via HTML5**
(e.g. HTML5 postMessage(...))

- Beispiel: Integration mit Bonita:
Kundenauswahl (situativ) verbessert
- Choose ruft *Bonita* auf

➔ *einen kleinen TEIL einer großen Anwendung mit vaadin umsetzen*



The screenshot shows the Bonitaopen solution interface. At the top left is the logo "bonitaopen solution". At the top right, it says "admin | Ausloggen" and "Bestandsauskunft". The main heading is "Auswahl via vaadin". Below this, there is a search filter area with "Von: 12.09.2013 11:26", "Bis:", and "Priorität: Normal". A modal dialog box is open, displaying information: "INFO: Insgesamt 5 Kunden.", "INFO: Prozess: Bestandsauskunft--1.0--1", and "Kundenliste:". The list contains the following data:

CUSTOMER NAME	CONTACT NAME
Rastl KG	Sepp Rastl
Rothbucher, Sebastian	Sebastian Rothbucher
Moes Bar	Moe Szislack
Test	Lasmiranda Densevilla
Skinner, Seymour	Seymour Skinner

At the bottom of the dialog is a "Choose" button.

- Klein, smart, situativ anfangen (können) – überschaubares Risiko
- weiteres Beispiel: Facebook-like „Wall“ für Kundenservice
(vgl. Projekt der MetLife, NYC:
<http://www.10gen.com/press/metlife-leapfrogs-insurance-industry-mongodb-powered-big-data-application>)
Verwendung von Vaadin (7), CouchDB, MVP, CDI, etc.
- Auswahl Standard-Posts in *Bonita*

u.v.a.m



The screenshot displays the 'bonitaopen solution' interface. At the top right, it shows 'admin | Ausloggen' and 'Bestandsauskunft'. The main heading is 'Eintrag auf Wall'. Below this, there is a metadata bar with 'Von: 12.09.2013 11:28', 'Bis:', and 'Priorität: Normal'. A dropdown menu for 'Standard-Post' is open, showing options: 'Lastschrift zurückholen' (selected), 'Auskunft zum Saldo - keine weitere Aktion', 'Chance auf X-Sell Anlage', and 'Lastschrift zurückholen'. The main content area shows a list of posts from 'Sebastian Rothbucher > Kundenservice Rechnung':
1. 'Nachfrage wg. Rabatt - an Kundenservice'
2. 'Hat angerufen und will Rabatt'
3. 'Sebastian Rothbucher > Kundenservice Rechnung: Bitte auch darum kümmern'
4. 'This is a second post on the customer wall'
5. 'Sebastian Rothbucher > Max Schlauberger: Max, sieh da mal rein!
6. 'This is a first post on the customer wall'
A 'Comments...' field is visible at the bottom of the post list. An 'Ende' button is located at the bottom right of the interface.

- **Was** sind tägliche Herausforderungen?
- **Warum** ist Vaadin bei deren Bewältigung hilfreich?
- **Wie** wird in Vaadin eine Anwendung entwickelt?
- **Womit** anfangen?



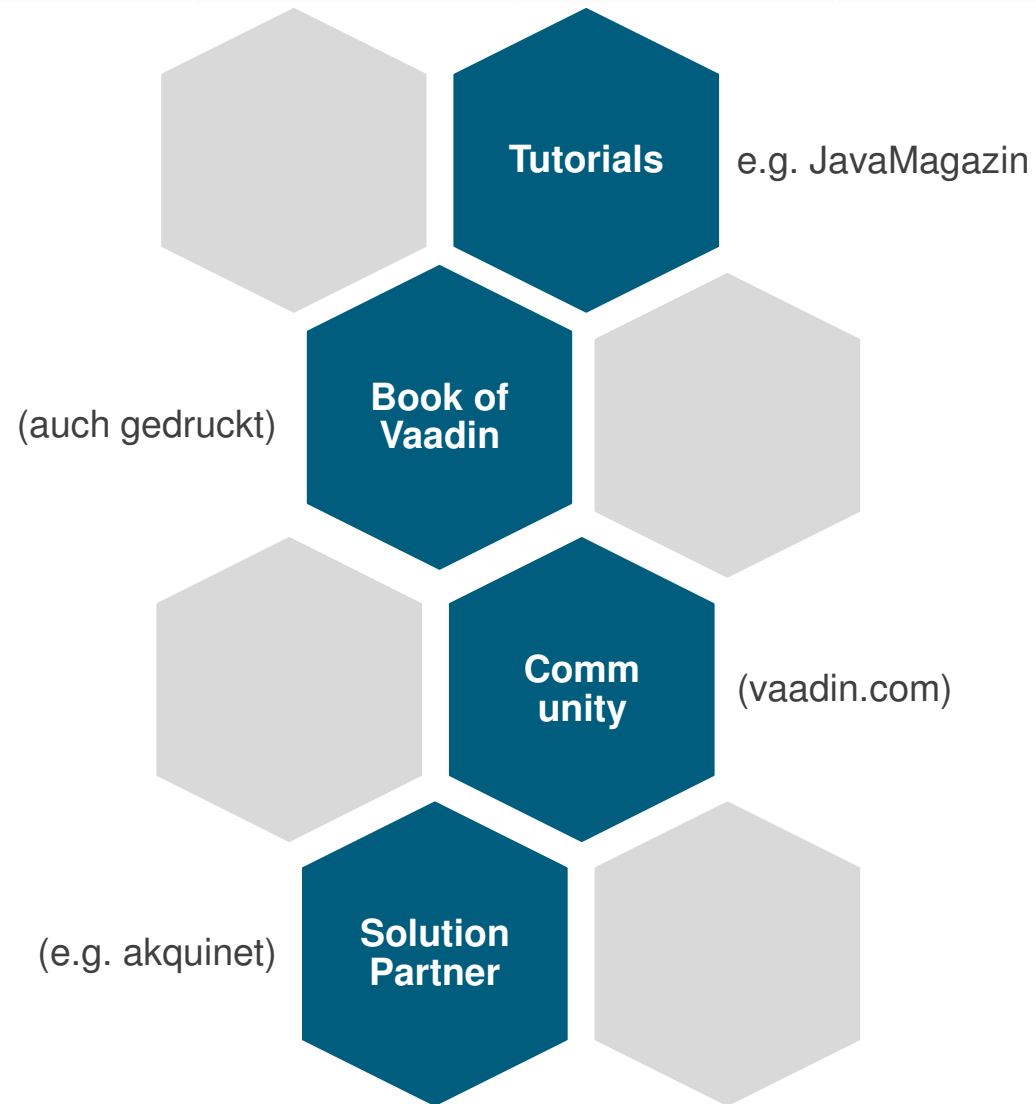
- **Loslegen!**

Leichter Einstieg
Schnell erste Ergebnisse

- **Achtung: Vaadin hilft, ist aber nur eine Technologie!**

Klare Anforderungen
Gute Projektmethodik
Permanentes Feedback
Sauberer Architektur

Wo fange ich an?



- **Peter Palchinsky (1875-1929) ist *Ihr Freund!***
- **Durchstich mit einem Use Case – einfach machen**
- **Muster etablieren (upfront, vorhersagbar) und wachsen**

- **Streng typisierte Web-Entwicklung**
- **Pure and Full Java**
- **Evolution Richtung Perfektion**
bzgl. Effizienz / Varianz / Akzeptanz
- *Legen Sie los!*



Wie können wir Sie unterstützen?



Ihre Ansprechpartner



Olaf Lange

***Geschäftsführer
akquinet engineering
GmbH***

*040-88173-2412
0179-121 42 06
olaf.lange@akquinet.de*



Sebastian Rothbucher

***Analyst / Developer / Berater
akquinet engineering
GmbH***

*0157-738 657 78
sebastian.rothbucher@akquinet.de*